National College of Ireland

Evaluating Machine Learning Models for Effective Phishing URL Detection

MSc Research Project
Cybersecurity

Reuel Mushayakarara
Student ID: 22244611

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Reuel Tafara Mushayakarara

Student ID: 22244611

Programme: MSc Cyber Security **Year:** Sep 2023

Module: Practicum 2

Supervisor: Michael Prior

Submission Due Date: 12 Aug 2024

Project Title: Evaluating Machine Learning Models for Effective Phishing URL Detection

Word Count: ..... **Page Count**.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Reuel Tafara Mushayakarara

Date: .....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluating Machine Learning Models for Effective Phishing URL Detection

Reuel Mushayakarara
22244611

Abstract

Phishing URL attacks deceive users into giving sensitive information by imitating legitimate URLs and this poses a cybersecurity threat. This research addresses the need for effective phishing URL detection by comparing various machine learning models. The focus is on determining which model among traditional (Logistic Regression), hybrid (Random Forest and Gradient Boosting Classifier) and advanced (Deep Neural Network) is most effective in detecting phishing URLs within a unified dataset. Another focus area is that of the impact of the feature extraction and selection to the performance of the models. Traditional models usually lack the capability to handle complex phishing URLs, while hybrid models offer better accuracy by combining multiple algorithms. Advanced models are complex to implement and need more computational resources. The study found that hybrid models had better accuracy and efficiency compared to other models. The practical implementation was through a web application that classifies a URL using the Random Forest model. The research further suggests enhancing the web application with user awareness training capabilities offering more effective and user-friendly phishing detection system.

1 Introduction

The ever-increasing reliance on digital platforms has created many opportunities for attackers to exploit the users with nefarious intentions. Phishing is a malicious cyber-attack of deceiving users into divulging sensitive personal information (Ripa, Islam, Arifuzzaman, 2021). This type of cyber-attack has grown in complexity and frequency which makes it a critical area of concern for cyber security. Being able to detect phishing URLs has become very important as a mechanism to protect users from such exploits. This research explores the use of machine learning models to enhance the accuracy and efficiency of phishing URL detection.

The research questions of this research are “Which machine learning models – traditional (Logistic Regression), hybrid (Random Forest and Gradient Boosting Classifier) or advanced (Deep Neural Network) – are most effective at accurately detecting phishing URLs within a unified dataset?” and “What is the impact of feature selection on the effectiveness of these models in detecting phishing URLs?”. The primary objective is to compare these models to assess their effectiveness within a unified dataset. It will also understand the role of feature extraction and selection on the performance of the models.

This research will contribute to the scientific literature by providing a comprehensive analysis of different machine learning models in phishing URL detection. It will shed more light into the performance of traditional, hybrid and advanced models based on the extracted and selected features. The findings will offer valuable information on into the design of the phishing detection system, focusing on the models and how best to extract the features.

The structure of the report is as follows: Section 2 focuses on the related works, Section 3 on the research methodology, Section 4 on the Design specification, Section 5 on the Implementation, Section 6 on the Evaluation and Section 7 on the Conclusion and Future works.

2 Related Work

Phishing URL detection is an important area of research in cybersecurity especially nowadays with many different ways to enhance accuracy and efficiency. This literature review will focus on previous works related to the use of machine learning models for phishing URL detection. This will emphasize their relevance to the Research Questions "Which machine learning models – traditional (Logistic Regression), hybrid (Random Forest & Gradient Boosting Classifier) or advanced (Deep Neural Network) – are most effective at accurately detecting phishing URLs within a unified dataset?" and "What is the impact of feature selection on the effectiveness of models in detecting phishing URLs? "

2.1 Traditional Machine Learning Models

The study by (Chiramdasu et al., 2021) aimed to investigate the effectiveness of the logistic regression model for detecting malicious URLs. It demonstrated how a relatively traditional model can be used to identify phishing URLs thereby protecting users from phishing cyber threats. The main results showed that logistic regression had the lowest testing time making it efficient for real-time applications. Limitations of the study point to lower accuracy scores compared to other models. This will result in the model not being able to be capable of catering to more sophisticated phishing URLs.

While the study by (Kumari et al., 2023) also made use of traditional machine learning models such as Random Forest, SVM, Logistic Regression and Naïve Bayes. The main focus was to prove how lexical and host-based features can effectively detect phishing URLs from legitimate ones. The results showed that Random Forest had the highest accuracy among the tested models. The major weakness is the inability to detect advanced phishing attempts that do not use lexical features.

The study (Ahammad et al., 2022) highlighted the use of Logistic Regression, Decision Trees and Support Vector Machines for phishing URL detection. The results showed that the light GBM model had the highest accuracy of 86% followed by Random Forest at 85.3%. This

confirms the robustness of traditional models even though limitations of handling complex phishing attacks still exist.

All three studies emphasize the effectiveness of traditional machine learning models in detecting phishing URLs. In terms of the accuracy metrics, Random Forest outperforms other models while logistic regression is very quick in testing. Light GBM showed the highest performance among traditional models. These traditional models set a baseline for comparing more complex approaches to handle the changing nature of phishing attacks.

2.2 Hybrid Machine Learning Models

The paper by (Jagdale and Chavan, 2022) focused on using a hybrid model consisting of multiple algorithms to improve detection accuracy. The hybrid model integrated models like Random Forest, SVM and Logistic Regression. The results of the study showed that the ensemble model performed better than the respective models in terms of accuracy and robustness. The main contribution was to show how a hybrid model leverages the strengths of the combined algorithms to enhance phishing detection. The limitations include increased computational requirements and longer times to do so.

The study by (Karim et al., 2023) also emphasizes the effectiveness of a hybrid approach to detect phishing URLs. The hybrid model achieved higher precision and recall scores compared to individual models. This showed the benefits of ensemble models in reducing false positives and negatives. One major challenge is the complexity when it comes to actual implementation.

Both papers show the superior performance of hybrid models over traditional single models. The ensemble method makes use of the combined strength resulting in improved accuracy and efficiency. The results suggest that hybrid models are effective in phishing detection and address some limitations of traditional models.

2.3 Advanced Machine Learning and Deep Learning Models

(Ripa et al., 2021) researched both traditional and advanced models such as XGBoost. The XGBoost model showed high accuracy and efficiency in detecting phishing URLs. This emphasizes the potential and capability of more advanced machine learning models techniques. The limitations of such models include increased computational requirements and increased complexity in model implementation.

(Aljabri and Mirza, 2022) investigated the use of deep learning models like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) alongside traditional ones. The deep learning models showed better performance in capturing complex patterns within the dataset. This resulted in higher accuracy compared to traditional models. A major limitation identified was that of the complexity in training and deploying deep learning models. It may not be feasible for real-time applications due to the great resource demands.

The study by (Dantwala et al., 2023) validated the effectiveness of advanced models. Different models were tested and the results showed Neural Network model achieving high

accuracy levels across all testing scenarios. This showed the advantages of advanced models in achieving higher accuracy and handling better complex phishing scenarios.

All three papers show that advanced machine learning and deep learning models perform better than traditional models in phishing URL detection. Models like XGBoost and Neural Networks can capture complex patterns and relationships within a dataset which leads to higher detection accuracy. The major drawback is the additional computational resources needed and it is more complex to implement.

2.4 Conclusion

See table 1 below for the summary of the Literature Review.

Study	Key Findings	Limitations
(Chiramdasu et al., 2021)	Efficient for real-time applications due to low testing time	Lower accuracy compared to other models
(Kumari et al., 2023)	Random Forest had the highest accuracy among tested model	Inability to detect advanced phishing attempts not using lexical features
(Ahammad et al., 2022)	Light GBM had highest accuracy (86%), followed by Random Forest (85.3%), confirming robustness of traditional models	Limitations in handling complex phishing attacks
(Jagdale and Chavan, 2022)	Ensemble model performed better in terms of accuracy and robustness	Increased computational requirements and longer processing times
(Karim et al., 2023)	Higher precision and recall scores, benefits in reducing false positives and negatives	Complexity in actual implementation
(Ripa et al., 2021)	XGBoost showed high accuracy and efficiency, showcasing the potential of advanced techniques	Increased computational requirements and complexity in implementation
(Aljabri and Mirza, 2022)	Deep learning models captured complex patterns, resulting in higher accuracy compared to traditional models	Complexity in training and deployment, resource-intensive, may not be feasible for real-time applications
(Dantwala et al., 2023)	Neural Networks achieved high accuracy across all testing scenarios, handling complex phishing scenarios better	Additional computational resources needed, more complex implementation

Table 1

Traditional machine learning models provide a baseline for phishing detection which offers simplicity and ease of interpretation without much complexity involved. Hybrid models then improve upon this by integrating multiple models which enhances the accuracy and

effectiveness of detection with added complexity. Finally the advanced models offer the highest accuracy with a trade-off of increased complexity, computational requirements and expertise to implement.

Research Niche

This literature review has highlighted the progression from traditional to hybrid and finally advanced machine learning models in phishing URL detection. Each of these methods offers unique strengths and weaknesses. The research will build on the identified gaps to evaluate the most effective models within a unified dataset.

3 Research Methodology

Dataset collection

The Dataset was obtained from the UC Irvine Machine learning repository under the CC BY 4.0 license (Prasad and Chandra, 2024). The license allows for the sharing and modification of the dataset for any purpose as long as it is correctly citated. The dataset contains 134 850 legitimate and 100 945 phishing URLs. The dataset contains 54 features extracted from the webpage source code and URL. This dataset is very important in the undertaking of the research to understands which traditional, hybrid or advanced models are effective

Data Processing

At this stage the dataset will undergo analysis and modelling steps to ensure that its clean, consistent and fit for feature extraction. The steps include the below.

- Data Integration – It involves combining data from different sources into a unified dataset. This can involve merging, joining many datasets.
- Data Reduction – This reduces how big the data is by selecting particular features, removing redundant data and applying reduction techniques.
- Data Cleaning – The data will undergo further manipulation to remove inaccurate, null and dealing with outliers to ensure the quality of the dataset.
- Data Transformation – At this step, the data is converted into a format that's easy to analyse. This may include changing categorical values into integer values.

Feature Engineering

This will involve the selection of the relevant features that will be used to train the different machine learning models. There are 54 features in this dataset and only a few of those will be selected in the context of the research. This is for the transformation and preparation of the data into a format that can be effectively used by the machine learning algorithms. The categorical features will be truncated from the dataset as the focus will be on the binary based type of features.

Training Data and Test Data

The cleaned and engineered dataset will be split into two groups, training and test data. The split will be done using the 80:20 ratio. The model will train with 80% of the dataset and use the 20% for the testing to evaluate its performance.

Learning Algorithm
Traditional, Hybrid and Advanced models will be chosen and at the end compare which models offer more efficient detection rates. Logistic Regression, Random Forest, Gradient Boosting Classifier and Deep Neural Network are the chosen models for this research

Train Model
The selected algorithm is trained using the 80% training data. During this process the model learns the patterns and relationships within the dataset to determine the accuracy and precision amongst other metrics.

Score Model
The remaining 20% of the data is used to make predictions based on the training data. It creates scores or prediction values for the test data assessing the model's performance on unseen data.

Evaluation Model
The algorithm is then evaluated using different metrics such as accuracy, precision, recall, F1-score. This helps in determining how effective and reliable the machine learning model is. These steps can be used in a cyclic process to improve the results of the model. This can be done by going back to refine the preceding steps like feature engineering or selecting different algorithms. This process can be repeated until the model meets the desired levels of performance.

4 Design Specification

The focus of this section will be on the design specification on how the research was conducted.

Logistic Regression is a machine learning model that is used for binary classification. It predicts the probability of an outcome to fall under one of the two classes i.e. Phishing or Legitimate, 0 or 1. It's used for scenarios where the outcome has two possible classes.

Random Forest is an ensemble machine learning model used for classification and regression. Ensemble learning involves the combination of multiple decision trees to improve the model performance. It uses bagging and random feature selection to create a different set of decision trees. It will then aggregate the final prediction from the predictions of all the trees.

Gradient Boosting is also an ensemble machine learning model. It builds models sequentially with each new model correcting the errors of the previous models. It uses gradient descent to reduce the loss function (errors) thereby improving the accuracy with the next model.

Deep Neural Network is an artificial neural network with multiple hidden layers between the input and output layers. It makes use of forward propagation and backpropagation to learn from the data and minimize the loss function.

Architecture

Fig 1 below depicts the architecture of the web URL Detector

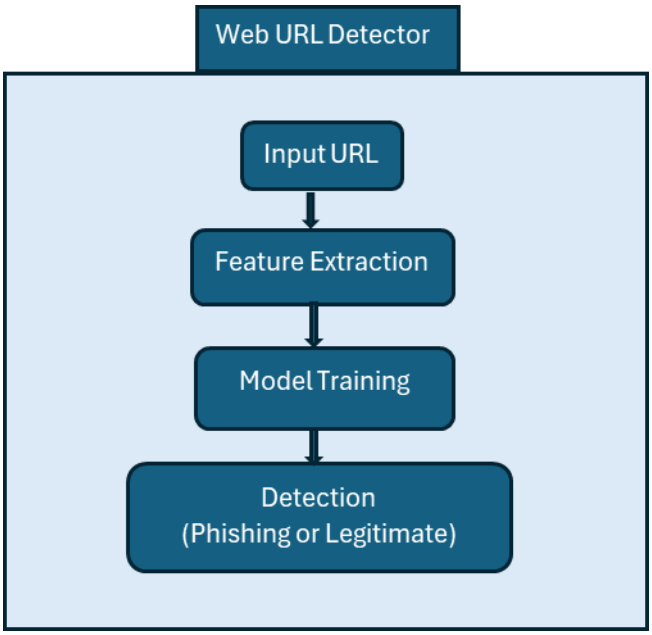

Fig 1

- Input – URL submitted by the user
- Process – The features will be extracted from the URL and passed to the trained model for the classification prediction.
- Output – The web application will then display whether the URL is Phishing or Legitimate along with the percentage of prediction.

5 Implementation

Dataset

The dataset (PhiUSIIL Phishing URL Dataset) used for the research was obtained from the UC Irvine Machine Learning Repository.

- It has a total of 235 795 instances of which 134 850 are legitimate and 100 945 are phishing URLs
- Legitimate URLs were collected from Open PageRank Initiative and phishing URLs from PhishTank, OpenPhish and MalwareWorld
- Contains 54 features
- Feature types are categorical (object type), integer (int64 type) and decimal (float64 type)
- Licensed under the CC BY 4.0 license. The license allows sharing and adaptation of the dataset for any purpose so long the appropriate credit is given.
- The dataset was created by Arvind Prasad and Shalini Chandra

Python Libraries

- Pandas – is a library used for data manipulation and analysis for Python (Pandas 2024). It handles structured data easily through data structures like Data frames. Pandas is critical for data analysis tasks and is used for reading, writing, data alignment and merging datasets.
- NumPy - is a fundamental Python package for scientific computing (NumPy 2024. It provides support for large multi-dimensional arrays, matrices and functions to operate on the arrays.
- Scikit-learn (sklearn) – is an open-source library that supports machine learning (Scikit-Learn 2024). It provides tools for data processing, model fitting, model selection and model evaluation. Sklearn.metrics includes functions for calculating accuracy, precision, recall, F1 score and confusion matrices. The metrices are then used to evaluate the effectiveness of machine learning algorithms. The sklearn.ensemble module provides ensemble-based machine learning algorithms such as Random Forest and Gradient Boosting classifiers. Ensemble models combine multiple individual models to create a single robust model with improved performance.
- Seaborn – is a Python data visualization library built on top of Matplotlib (Waskom, 2021). It offers an interface for drawing statistical graphics that are informative and attractive. Seaborn is best suited for visualizing variable relationships and datasets.
- Matplotlib.pyplot – is a plotting library used in Python to create different types of visualizations i.e. static or animated (Matplotlib 2024). It's usually used to create plots, bar charts, histograms and pie charts. Matplotlib is very customizable and generally used for data visualization in machine learning.

Machine learning algorithms

The following machine learning algorithms were used in this research

- Logistic Regression
- Random Forest
- Gradient Boost
- Deep Neural Network

Feature Engineering

This is a very important step in machine learning that includes the extraction and selection of features (Verma and Chandra, 2023).

For experiment 1 the f features on the original dataset will be used

- TLD – The Top Level Domain (TLD) of each URL was identified and it was observed that phishing URLs use uncommon TLDs.
- URLLength – the length of each URL was measured and the phishing URLs are longer.
- IsDomainIP – checked if the URL used an ip address instead of a domain name.
- NoOfSubDomain – checked the number of subdomains. Phishing URLs usually use multiple subdomains.
- NoOfObfuscatedChar – checked the obfuscated characters in the URL

- IsHTTPS – checked if the URL used http or https. Legitimate URLs use the secure HTTPS.
- No. of digits, equal, qmark, amp – counted the number of digits and special characters in the URL.
- LargestLineLength – the length of the longest line in the HTML code was measured and this indicated potential obfuscation.
- HasTitle – verified if a title tag was present.
- HasFavicon – checked for the precense of a favicon tag.
- IsResponsive – verified if there was a response from the website.
- NoofURLRedirect – counted the number of redirects in the HTML code
- HasDescription – checked for a description meta tag.
- NoOfPopup, NoOfiFrame – counted the number of popups and iframes.
- HasExternal FormSubmit – checked if the forms sent data to external URLs.
- HasCopyrightInfo, HasSocialNet – the presence of copyright and social networking information was verified.
- HasPasswordField, HasSubmitButton – the presence of password fields and submit buttons was checked.
- HasHiddenFields – They looked for hidden fields in the HTML code
- Bank, Pay, Crypto – checked to see if the webpage was asking for any payment details.
- NoOfImage – The number of images on the webpage was counted.
- NoOfJS – checked how many Java scripts were embedded in a webpage.
- No)fSelfRef, NoOfEmptyRef, NoOfExternalRef – the number of hyperlinks navigating to itself, empty links and external links were counted.
- CharContinuationRate – measured the rate of continuation of alphabets, digits and special characters in the URL
- URLTitleMatchScore – the match score between the URL and webpage title was calculated. A lower score indicated a phishing URL.
- URLCharProb – analyzed the probability of each character in the URL based on patterns from legitimate and phishing URLs.
- TLDLegitimateProb – the probability of the TLD being legitimate was calculated based on its occurrence in the top 10 million websites.

Further to these features already found on the dataset, new features were then created for experiment 2. These include

- url_len – extracted the length of the url.
- letters_count – count on letters in the url.
- digits_count – count of digits in the url.
- special_chars_count – count of special characters.
- Shortened – checked if any shortened service was in use
- abnormal_url – checked if the url was abnormal
- secure_http – checked if it was using https
- have_ip – checked if an ip address was used

For the experiment 3, 30 new features were extracted from the URL and are as below

- UsingIp – checks if the URL is an ip address
- longUrl – checks the length of the URL
- shortUrl – checks if the URL is using any known shortening services
- symbol – checks for the @ symbol within the URL
- redirecting – it checks for any redirection within the URL
- prefixSuffix – checks if the domain has the "–" sign
- SubDomains – counts the number of the dots in the URL
- Https – checks if the URL uses https
- DomainRegLen – checks the domain registration length using the whois data
- Favicon – checks the favicons URL
- NonStdPort -checks for non-standard ports in the domain
- HTTPSDomainURL – checks if the domain has https
- RequestURL – checks for images, audio, embeds, iframes within the source URL
- AnchorURL – checks the safety of anchor URLs
- LinksinScriptTags – checks the links within the script tags
- ServerFormHandler – checks the action of the forms within the URL
- InfoEmail – looks for any mailto links
- AbnormalURL – checks the URL with whois data
- WebsiteForwarding – looks for any redirections
- StatusBarCust – checks for scripts that alter the status bar
- DisableRightClick- checks for scripts that disable right click
- UsingPopupWindow – checks for any alert calls in the URL
- IframeRedirection – looks for any iframe tag redirection
- AgeofDomain – calculates the age of the domain from the whois data
- DNSRecording – this checks the age of the DNS records
- WebsiteTraffic – checks the ranking of the URL according to Alexa.com
- PageRank – checks the ranking of the URL page using checkpagerank service
- GoogleIndex – checks if the URL is indexed by Google
- LinksPointingToPage – checks for the number of links pointing to the page
- StatsReport – checks for known suspicious URLs and ip addresses

Model Training

The dataset for this research was split using the 80:20 ratio. The training of the algorithms was done on 80% of the dataset and the testing was done on the remaining 20%

Web URL Detector Components

- Retrain_model.py – the dataset used for training the model is loaded from a csv file. For experiment 2, the dataset (newdataset1.csv) contained the 8 extracted features. For experiment 3, the dataset (newdatset.csv) contain the 30 extracted features. The data is split into training and testing groups using an 80/20 ratio.

A Random Forest model is used to train the model. The trained model is then saved as "model.pkl" using the pickle library

- Feature.py – the extraction of the features is carried out. These features are then used for the purpose of classifying the URL
- App.py – a Flask web application is set up to serve as the interface for detecting the phishing URLs. When a user inputs a URL, the application will extract features using the feature.py. The extracted features are then passed to the trained model for the prediction whether the URL is phishing or legitimate. The result including the probability percentage is then displayed to the user.

6 Evaluation

Multiple experiments were conducted so as to determine the performances of the models under different models. The performances of the models under review will be evaluated using the metrices like accuracy, f1 score, precision, recall, confusion matrix.

6.1 Experiment 1

The models were trained using the original features from the dataset.

Logistic Regression – the model achieved an Accuracy of 0.9999, F1-score of 0.9999, Precision of 0.9999 and Recall of 1. The confusion matrix indicated 4 false negatives

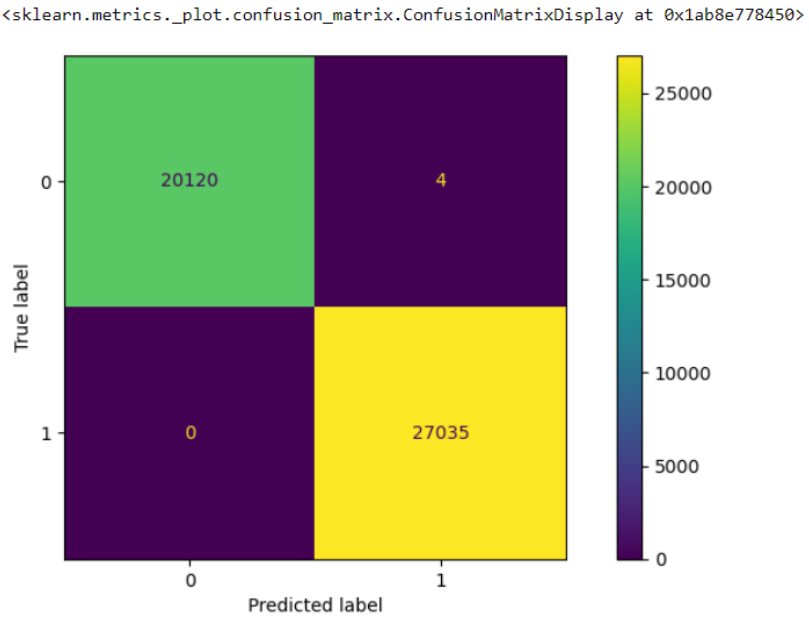

Fig 2

Random Forest – the model achieved perfect results with an Accuracy of 1, F1-score of 1, Precision of 1 and Recall of 1. The confusion matrix had no false positives or negatives.

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1d0507d1450
>

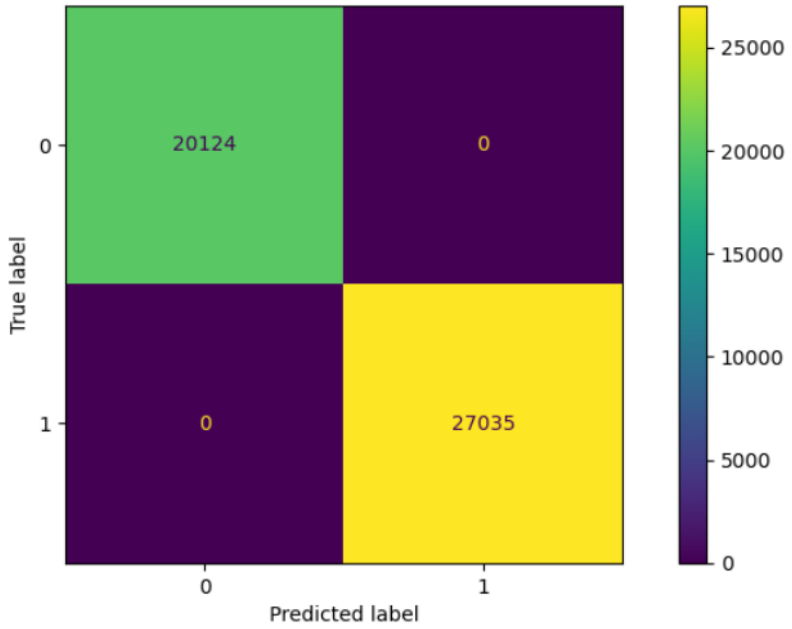

Fig 3

Gradient Boosting Classifier – the model achieved perfect results with an Accuracy of 1, F1-score of 1, Precision of 1, Recall of 1. The confusion matrix showed no false positives or negatives

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1d051d39050
>

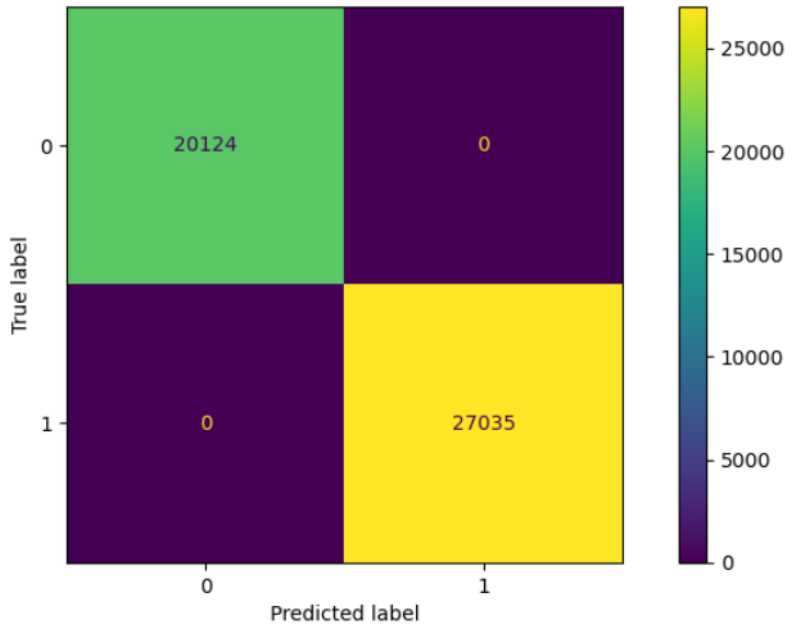

Fig 4

Deep Neural Network – the model achieved an Accuracy of 0.9999, F1-score of 0.9999, Precision of 0.9999, Recall of 0.9999. the confusion matrix had 2 false positives and 2 false negatives.

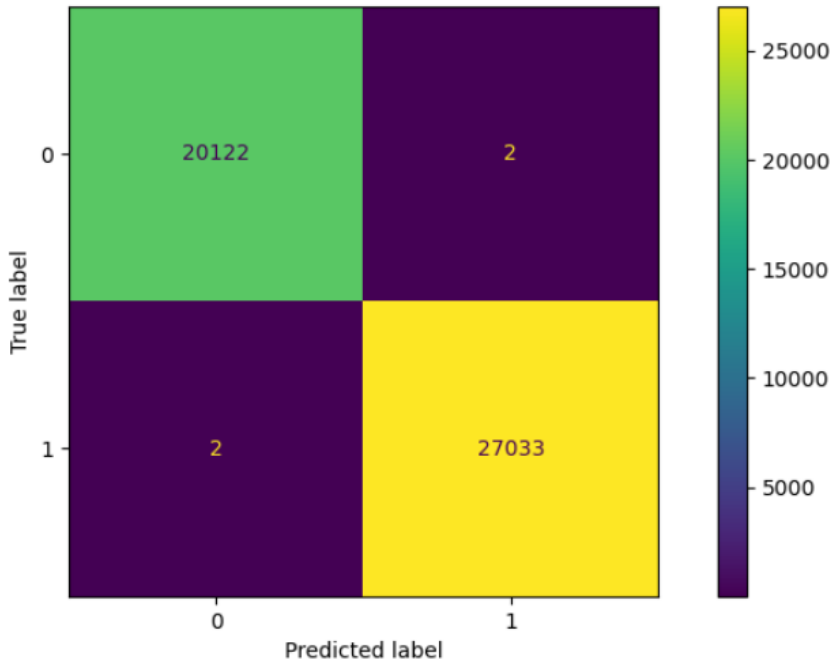

Fig 5

Overally, the experiment indicates that Random Forest and Gradient Boosting Classifier were both effective and achieved 100% classification without any errors. Logistic Regression and Deep Neural Network both achieved high levels of effectiveness although with slight errors.

The results show that hybrid models were capable of accurately detecting phishing URLs. The almost perfect results from all models show that the original features were informative for the classification task. This is typical within theoretical scenarios. Hence the need for more experiments with different features to see what impact it will have on the effectiveness of the models.

6.2 Experiment 2

The results below are from running the models using 8 features extracted from the URL column of the dataset.

Logistic Regression – achieved an Accuracy of 0.9941, F1-score of 0.9949, Precision of 0.9903 and Recall of 0.9994. The confusion matrix identified 264 false positives and 15 false negatives

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x29f16a8c050
>

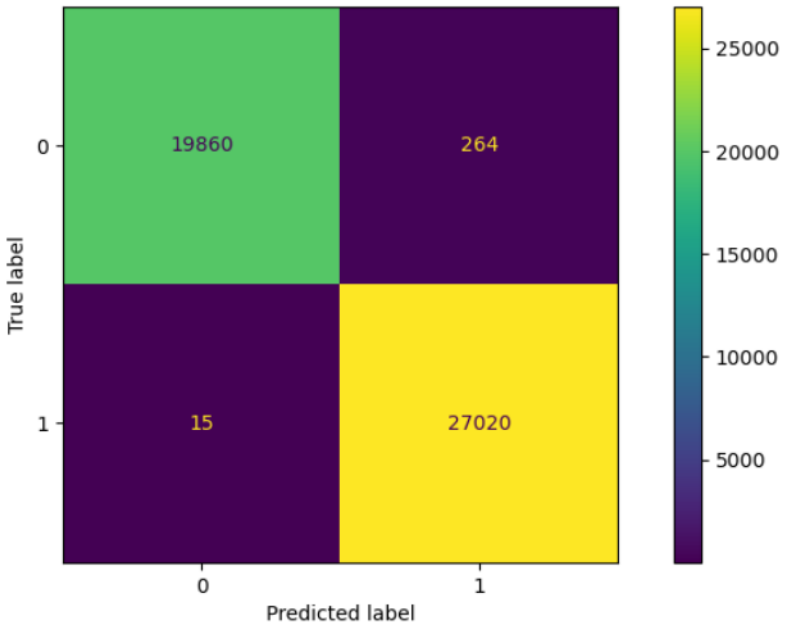

Fig 6

Random Forest – achieved an Accuracy of 0.9951, F1-score of 0.9958, Precision of 0.9928 and Recall of 0.9987. The Confusion Matrix identified 196 false positives and 34 false negatives.

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x29f22d3bf90
>

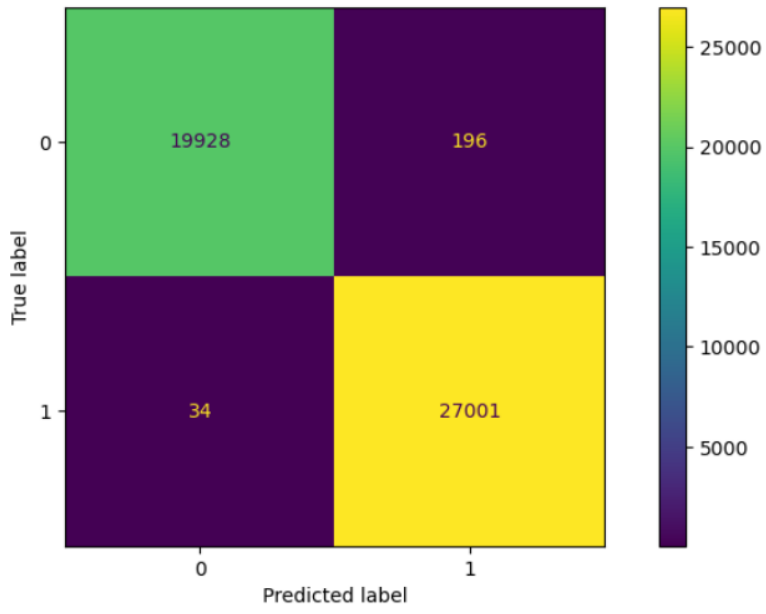

Fig 7

Gradient Boosting – achieved an Accuracy of 0.9885, F1-score of 0.9900, Precision of 0.9958 and Recall of 0.9943. The confusion matrix identified 388 false positives and 153 false negatives.

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x29f237a6390
>

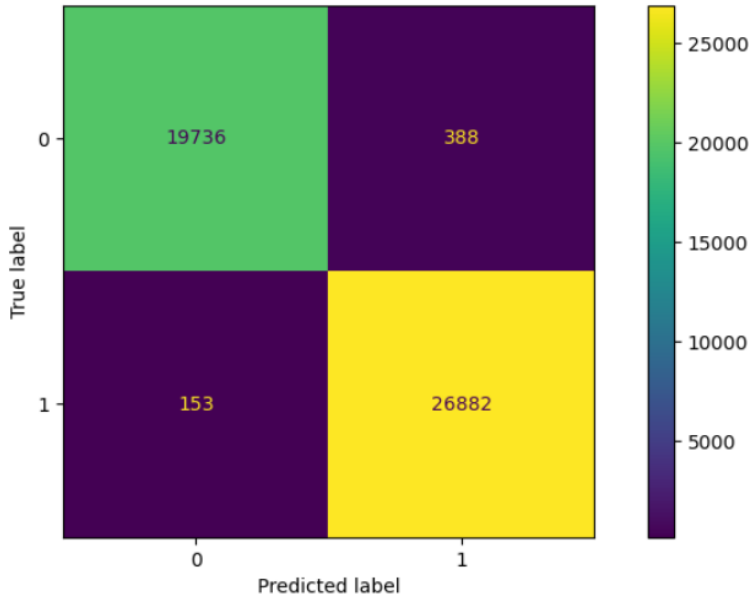

Fig 8

Deep Neural Network – achieved an Accuracy of 0.9948, F1-score 0f 0.9955, Precision of 0.9924 and Recall of 0.9987. The confusion matrix identified 208 false negatives and 36 false negatives

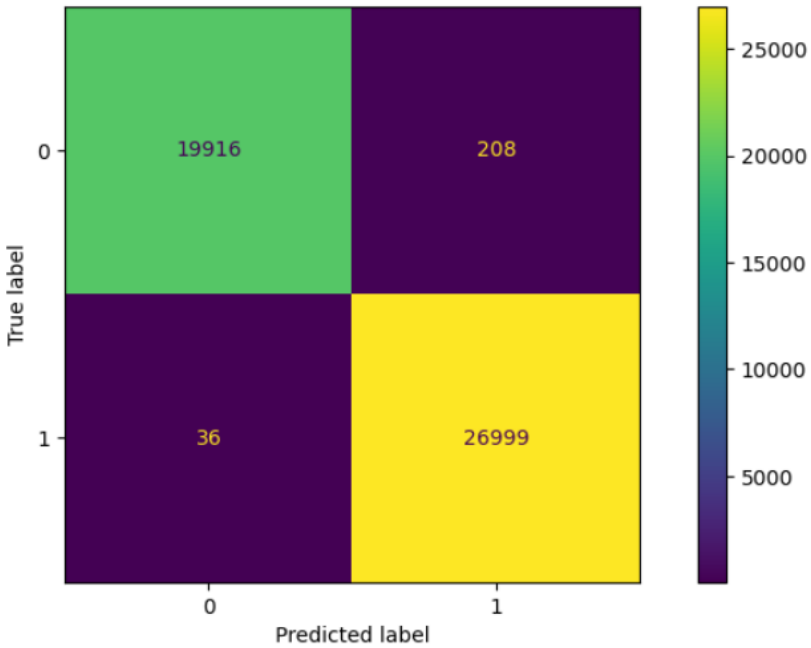

Fig 9

Artifact Screenshot

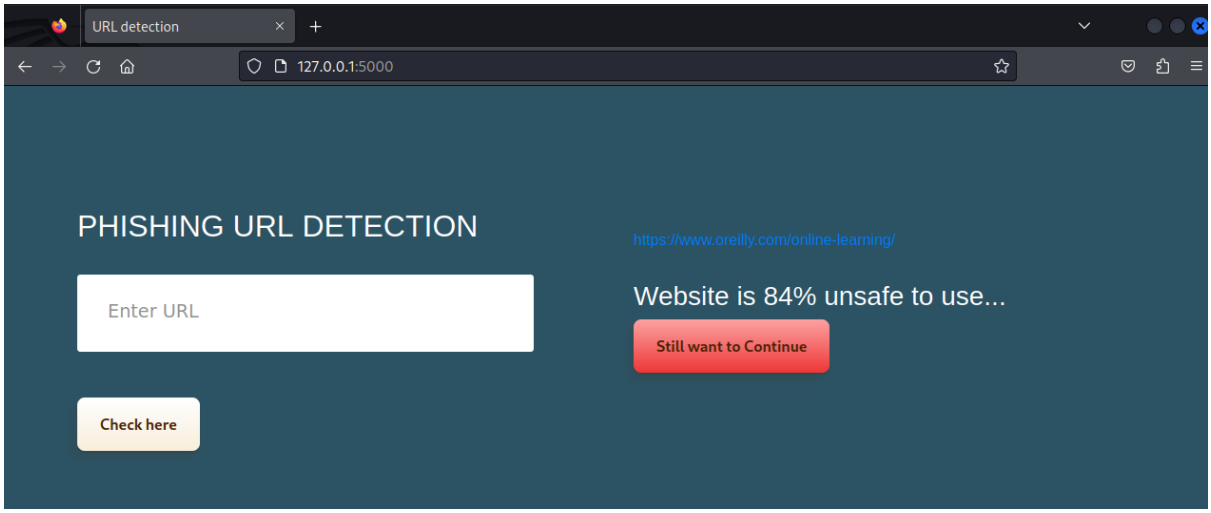

Fig 10

The screenshot above shows the actual implementation of experiment 2 using Random Forest as the model. It shows experiment 2 failing to correctly identify an URL despite it having a higher accuracy, precision levels.

Overally the results indicate that reducing the number of features to 8 led to a decrease in the performance of the models. This impact was most experienced by the Gradient Boosting classifier. The results also point towards the fact that 8 features may not be enough to capture all the information needed for a correct classification.

6.3 Experiment 3

The results below are from running the models with 30 features extracted from the URL column.

Logistic Regression – achieved an Accuracy of 0.8761, F1-score of 0.8964, Precision of 0.9330, Recall of 0.8626. The confusion matrix showed 4018 false positives and 1813 false negatives

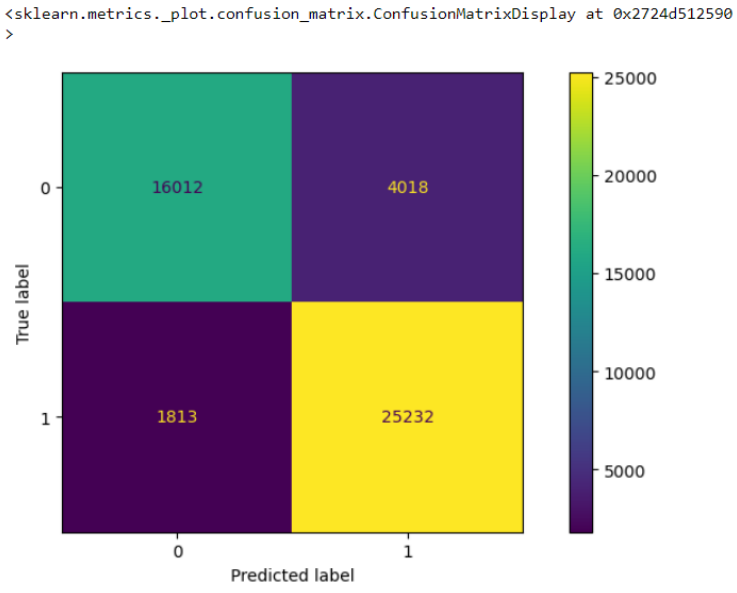

Fig 11

Random Forest - achieved an Accuracy of 0.8977, F1-score of 0.9123, Precision of 0.8987 and Recall of 0.9263. The confusion matrix showed 2824 false positives and 1994 false negatives

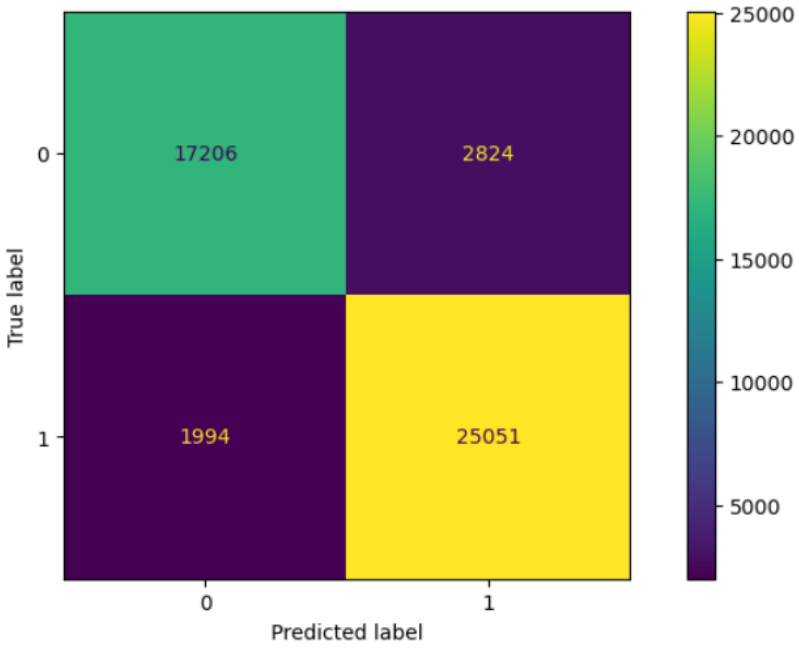

Fig 12

Gradient Boosting Classifier – achieved an Accuracy of 0.8976, F1-score of 0.9123, Precision of 0.8987 and Recall of 0.9263. The confusion matrix showed 2825 false positives and 1994 false negatives

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2724759a9d0
>
```

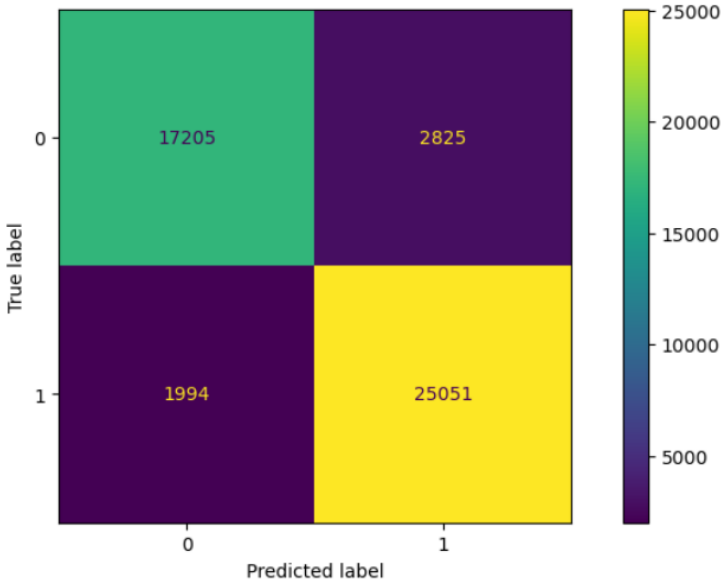

Fig 13

Deep Neural Network – achieved an Accuracy of 0.8974, F1-score of 0.9121, Precision of 0.8983 and Recall of 0.9263. The confusion matrix showed 2837 false positives and 1884 false negatives

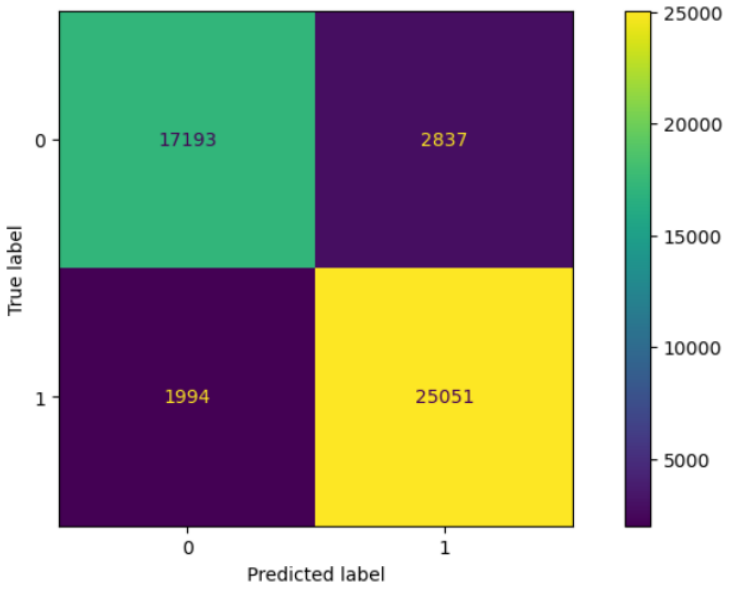

Fig 14

Artifact Screenshot

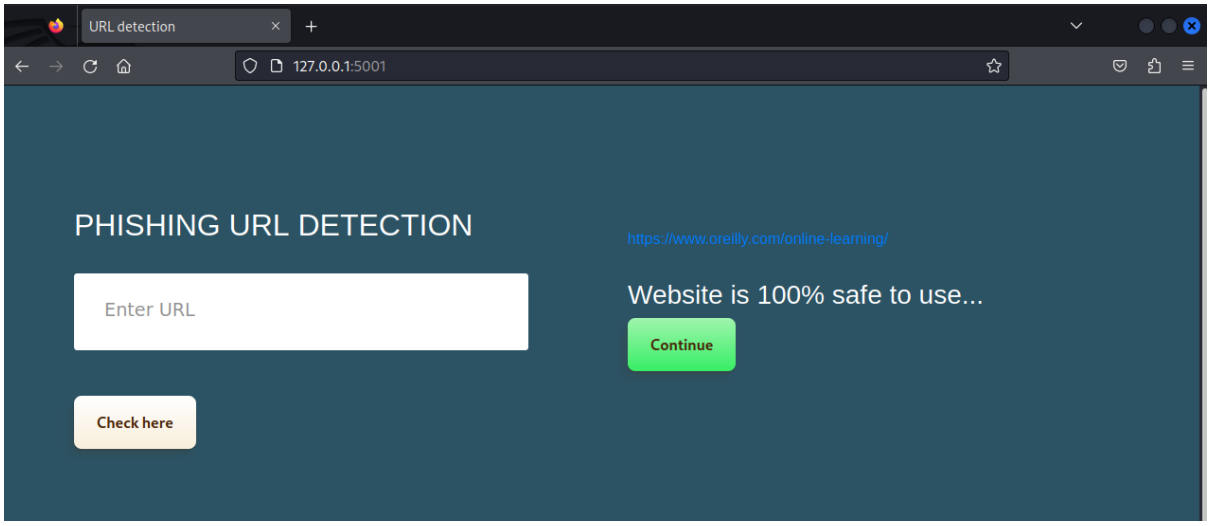

Fig 15

The screenshot above shows the actual implementation of experiment 3 using Random Forest as the model. Here the URL was successfully identified whereas the same URL wasn't in Experiment 2. This was done even with lower performance metrics compared to experiment 2. This shows that features that added noise in the actual training and testing, are essential in actually detecting URLs in real-time.

6.4 Discussion

The experiments show the different effective levels in detecting phishing URLs. The original dataset (experiment 1) produced the perfect results across all models which is possible theoretically but not in practical terms. Experiment 2 running with 8 features resulted in a slight decrease in the overall metrics for all models. This highlighted limitations in capturing enough details to accurately classify URLs. This was actually revealed when experiment 2

failed to correctly identify a URL. Experiment 3 had overall less accurate results but it managed to correctly identify the same URL used in experiment 2. This is another indicator that additional features introduced noise resulting in lower efficacy of the models. At the same time the additional features provided additional information which enabled the correct classification of the URL.

The research effectively compared traditional, hybrid and advanced machine learning models. One key point based on the 3 experiments is around the importance of feature extraction and selection. Feature extraction and selection plays a critical role in determining the model performance and ability to predict correctly. The way of extracting features can be improved to focus on the most relevant ones without adding unnecessary noise. This can be included as part of the future research to focus on how to optimize feature extraction and selection. This will also include analyzing which feature is important and applying reduction techniques.

The findings from this research align with the previous research in showing that hybrid (ensemble) models are highly effective in phishing URL detection. The mixed results from the Deep Neural Network model highlight the challenges of feature extraction and selection in neural network training. The research also shows that a more advanced model is not always better than traditional or hybrid models. It also shows how feature selection can significantly affect the performance of a model.

This research adds to the understanding of how different machine learning models and feature grouping impact phishing detection. The lessons can be applied to enhance phishing URL systems through knowing how to select the correct features and models for the actual implementation. The study suggests an approach of adding features which aim for a balance between complexity and interpretability.

The comprehensive evaluation of the three experiments shows the important role of feature selection in the effective detection of phishing URLs. While advanced models offer more capability and functionality, hybrid ensemble models offer the best performance especially with well selected features.

7 Conclusion and Future Work

The primary objective of this research was to determine which machine learning models – traditional, hybrid or advanced – are most effective at accurately detecting phishing URLs within a unified dataset. The research also aimed at finding out the impact of feature selection on the performance of the models. The research successfully answered the questions by comparing different models namely Logistic Regression, Random Forest, Gradient Boosting Classifier and Deep Neural Network.

The key findings from the experiments show that hybrid models Random Forest and Gradient Boosting Classifier had better results than the traditional and advanced models. The research also revealed that advanced models like Deep Neural Network have more capabilities, functionality and require more computational resources. The advanced models may not always produce better results when compared to simpler models. Another key finding was that of how important feature extraction and selection is to the performance of the models. The inclusion of irrelevant features can introduce noise which then affects the results and

vice versa is true. This was the case with the differences between the results from experiment 2 and 3. Experiment 2 had better results but when it came to the actual implementation, it failed to classify the URL correctly. While experiment 3 had poor results but it managed to correctly classify the URL.

The implications of this research are significant for the field of phishing detection. The results show the importance of selecting the right features and the potential of hybrid models in creating effective and accurate detecting systems. However, the limitations include the potential overfitting of models due to the high accuracy results ideal in theoretical and not practical settings. This was evidenced from the practical implementation of experiment 2 and 3.

Future Works

The future works could explore the enhancement of the web application to also provide user awareness tips for the users. In addition to detecting the phishing URLs, the web application can be used to educate users in identifying and avoiding phishing attempts. This can be done through incorporating the below features and functionalities

- Daily tips and alerts – provide users with daily tips and alerts on how to identify phishing attempts.
- Interactive quizzes – short quizzes can be included to gauge the user's knowledge of different phishing methods. This can be a way to keep users engaged and to reinforce the lessons learnt.

This dual functionality or capability will enhance the value of the web URL phishing detector application.

References

Ahammad, S.H., Kale, S.D., Upadhye, G.D., Pande, S.D., Babu, E.V., Dhumane, A.V., Bahadur, Mr.D.K.J., 2022. Phishing URL detection using machine learning methods. *Adv. Eng. Softw*are. 173, 103288. https://doi.org/10.1016/j.advengsoft.2022.103288

Aljabri, M., Mirza, S., 2022. Phishing Attacks Detection using Machine Learning and Deep Learning Models, 2022 7th *International Conference on Data Science and Machine Learning Applications (CDMA)*.pp. 175–180. https://doi.org/10.1109/CDMA54072.2022.00034

Chiramdasu, R., Srivastava, G., Bhattacharya, S., Reddy, P.K., Reddy Gadekallu, T., 2021. Malicious URL Detection using Logistic Regression, 2021 *IEEE International Conference on Omni-Layer Intelligent Systems (COINS).* pp. 1–6. https://doi.org/10.1109/COINS51742.2021.9524269

Dantwala, V., Lakhani, R., Shekokar, N., 2023. A Novel Technique to Detect URL Phishing based on Feature Count. 2023 3rd *International Conference on Intelligent Communication and Computational Techniques (ICCT).* pp. 1–5. https://doi.org/10.1109/ICCT56969.2023.10075943

Jagdale, N., Chavan, P., 2022. Hybrid Ensemble Machine Learning Approach for URL Phishing Detection. 2022 2nd *Asian Conference on Innovation in Technology (ASIANCON)*. pp. 1–8. https://doi.org/10.1109/ASIANCON55314.2022.9908667

Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S.B., Joga, S.R.K., 2023. Phishing Detection System Through Hybrid Machine Learning Based on URL. *IEEE Access* 11, 36805–36822. https://doi.org/10.1109/ACCESS.2023.3252366

Kumari, M.S., Priya, C.K., Bhavya, G., Neha, H., Awasthi, M., Tripathi, S., 2023. Viable Detection of URL Phishing using Machine Learning Approach. *E3S Web Conf.* 430, 01037. https://doi.org/10.1051/e3sconf/202343001037

Matplotlib (2024) Matplotlib — Visualization with Python. Available at: https://matplotlib.org/ [Accessed 1.6.24].

NumPy (2024) *NumPy*. Available at: https://numpy.org/ [Accessed 1.6.24].

Pandas (2024) *pandas - Python Data Analysis Library* Available at: https://pandas.pydata.org/ [Accessed 1.6.24].

Prasad, A., Chandra, S., 2024. PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. *Computer. Security*. 136, 103545. https://doi.org/10.1016/j.cose.2023.103545

Ripa, S.P., Islam, F., Arifuzzaman, M., 2021. The Emergence Threat of Phishing Attack and The Detection Techniques Using Machine Learning Models, 2021 *International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI).* pp. 1–6. https://doi.org/10.1109/ACMI53878.2021.9528204

Verma, R., Chandra, S., 2023. RepuTE: A soft voting ensemble learning framework for reputation-based attack detection in fog-IoT milieu. *Eng. Appl. Artif. Intell.* 118, 105670. https://doi.org/10.1016/j.engappai.2022.105670

Waskom, M., 2021. *seaborn: statistical data visualization*. J. Open Source Software. https://doi.org/10.21105/joss.03021